TÀI LIỆU ÔN THI VÀO LỚP 10 THPT – MÔN TIN HỌC

PHẦN I: LÝ THUYẾT

CHƯƠNG I: CÁC THÀNH PHẦN CƠ BẢN CỦA NNLT PASCAL

1. Các bước cơ bản khi lập một chương trình Pascal

Bước 1: Soạn thảo chương trình.

Bước 2: Dịch chương trình (nhấn phím **F9**), nếu có lỗi thì phải sửa lỗi.

Bước 3: Chạy chương trình (nhấn phím **Ctrl-F9**).

2. Cấu trúc chung của một chương trình Pascal

{ *Phần tiêu đề* }

PROGRAM Tên_chương_trình;

{ *Phần khai báo* }

USES.....;

CONST.....;

TYPE.....;

VAR.....;

PROCEDURE.....;

FUNCTION.....;

.....

{ *Phần thân chương trình* }

BEGIN

.....

END.

Ví dụ 1: Chương trình Pascal đơn giản nhất

```
BEGIN
  Write('Hello World!');
END.
```

Ví dụ 2:

```
Program Vidu2;
Const PI=3.14;
Var R,S:Real;
Begin
  R:=10;{Bán kính đường tròn}
  S:=R*R*PI;{Diện tích hình tròn}
  Writeln('Dien tich hinh tron
= ', S:0:2); { In ra màn hình }
  Readln;
End.
```

3. Các thành phần cơ bản của ngôn ngữ Pascal

3.1. Từ khóa

Từ khoá là các từ mà Pascal dành riêng để phục vụ cho mục đích của nó. (Chẳng hạn như: **BEGIN, END, IF, WHILE,...**)

Chú ý: *Với Turbo Pascal 7.0 trở lên, các từ khoá trong chương trình sẽ được hiển thị khác màu với các từ khác*.

3.2. Tên (định danh)

Định danh là một dãy ký tự dùng để đặt tên cho các hằng, biến, kiểu, tên chương trình con... Khi đặt tên, ta phải chú ý một số điểm sau:

- Không được đặt trùng tên với từ khoá
- Ký tự đầu tiên của tên không được bắt đầu bởi các ký tự đặc biệt hoặc chữ số.
- Không được đặt tên với ký tự space,các phép toán.

Ví dụ: Các tên viết như sau là sai

1XYZ Sai vì bắt đầu bằng chữ số.

#LONG Sai vì bắt đầu bằng ký tự đặc biệt.

FOR Sai vì trùng với từ khoá.

KY TU Sai vì có khoảng trắng (space).

LAP-TRINH Sai vì dấu trừ (-) là phép toán.

3.3. Dấu chấm phẩy (;)

Dấu chấm phẩy được dùng để ngăn cách giữa các câu lệnh. Không nên hiểu dấu chấm phẩy là dấu kết thúc câu lệnh.

Ví dụ:

FOR i:=1 TO 10 DO Write(i);

Trong câu lệnh trên, lệnh Write(i) được thực hiện 10 lần. Nếu hiểu dấu chấm phẩy là kết thúc câu lệnh thì lệnh Write(i) chỉ thực hiện 1 lần.

3.4. Lời giải thích

Các lời bàn luận, lời chú thích có thể đưa vào bất kỳ chỗ nào trong chương trình để cho người đọc dễ hiểu mà không làm ảnh hưởng đến các phần khác trong chương trình. Lời giải thích được đặt giữa hai dấu ngoạc { và } hoặc giữa cụm dấu (* và *).

Ví dụ:

Var a,b,c:Rea; **{***Khai báo biến***}**

Delta := b*b – 4*a*c; **(*** *Tính delta để giải phương trình bậc 2* ***)**

BÀI TẬP THỰC HÀNH

1. Khởi động Turbo Pascal.
2. Nhập vào đoạn chương trình sau:

```
Uses Crt;
Begin
  Writeln('************************************************');
  Writeln('* CHUONG TRINH PASCAL DAU TIEN CUA TOI *');
  Writeln('*        Oi! Tuyet voi!... *);
  Writeln('************************************************');
  Readln;
End.
```

CHƯƠNG 2: CÁC KIỂU DỮ LIỆU CƠ BẢN
KHAI BÁO HẰNG, BIẾN, KIỂU, BIỂU THỨC VÀ CÂU LỆNH

I. CÁC KIỂU DỮ LIỆU CƠ BẢN

1. Kiểu logic

- Từ khóa: **BOOLEAN**
- Miền giá trị: **(TRUE, FALSE)**.
- Các phép toán: phép so sánh (=, <, >) và các phép toán logic: AND, OR, XOR, NOT.

Trong Pascal, khi so sánh các giá trị boolean ta tuân theo qui tắc: FALSE < TRUE.

Giả sử A và B là hai giá trị kiểu Boolean. Kết quả của các phép toán được thể hiện qua bảng dưới đây:

A	B	A AND B	A OR B	A XOR B	NOT A
TRUE	TRUE	TRUE	TRUE	FALSE	FALSE
TRUE	FALSE	FALSE	TRUE	TRUE	FALSE
FALSE	TRUE	FALSE	TRUE	TRUE	TRUE
FALSE	FALSE	FALSE	FALSE	FALSE	TRUE

2. Kiểu số nguyên

2.1. Các kiểu số nguyên

Tên kiểu	Phạm vi	Dung lượng
Shortint	-128 → 127	1 byte

Byte	$0 \rightarrow 255$	1 byte
Integer	$-32768 \rightarrow 32767$	2 byte
Word	$0 \rightarrow 65535$	2 byte
LongInt	$-2147483648 \rightarrow 2147483647$	4 byte

2.2. Các phép toán trên kiểu số nguyên

2.2.1. Các phép toán số học:

+, -, *, / (phép chia cho ra kết quả là số thực).

Phép chia lấy phần nguyên: **DIV** (Ví dụ : 34 DIV 5 = 6).

Phép chia lấy số dư: **MOD** (Ví dụ: 34 MOD 5 = 4).

2.2.2. Các phép toán xử lý bit:

Trên các kiểu ShortInt, Integer, Byte, Word có các phép toán:

- NOT, AND, OR, XOR.

A	B	A AND B	A OR B	A XOR B	NOT A
1	1	1	1	0	0
1	0	0	1	1	0
0	1	0	1	1	1
0	0	0	0	0	1

- SHL (phép dịch trái): a SHL n $\Leftrightarrow a \times 2^n$
- SHR (phép dịch phải): a SHR n $\Leftrightarrow$ a DIV 2^n

3. Kiểu số thực

3.1. Các kiểu số thực:

Tên kiểu	Phạm vi	Dung lượng
Single	$1.5\times10^{-45} \rightarrow 3.4\times10^{+38}$	4 byte
Real	$2.9\times10^{-39} \rightarrow 1.7\times10^{+38}$	6 byte
Double	$5.0\times10^{-324} \rightarrow 1.7\times10^{+308}$	8 byte
Extended	$3.4\times10^{-4932} \rightarrow 1.1\times10^{+4932}$	10 byte

Chú ý: Các kiểu số thực Single, Double và Extended yêu cầu phải sử dụng chung với bộ đồng xử lý số hoặc phải biên dịch chương trình với chỉ thị **{$N+}** để liên kết bộ giả lập số.

3.2. Các phép toán trên kiểu số thực: **+, -, *, /**

Chú ý: *Trên kiểu số thực không tồn tại các phép toán DIV và MOD.*

3.3. Các hàm số học sử dụng cho kiểu số nguyên và số thực:

SQR(x): Trả về x^2

SQRT(x): Trả về căn bậc hai của x ($x \geq 0$)

ABS(x): Trả về |x|

SIN(x): Trả về sin(x) theo radian

COS(x): Trả về cos(x) theo radian

ARCTAN(x): Trả về arctang(x) theo radian

LN(x): Trả về ln(x)

EXP(x): Trả về e^x

TRUNC(x): Trả về số nguyên gần với x nhất nhưng bé hơn x.

INT(x): Trả về phần nguyên của x

FRAC(x): Trả về phần thập phân của x

ROUND(x): Làm tròn số nguyên x

PRED(n): Trả về giá trị đứng trước n

SUCC(n): Trả về giá trị đứng sau n

ODD(n): Cho giá trị TRUE nếu n là số lẻ.

INC(n): Tăng n thêm 1 đơn vị (n:=n+1).

DEC(n): Giảm n đi 1 đơn vị (n:=n-1).

4. Kiểu ký tự

- Từ khoá: **CHAR**.
- Kích thước: 1 byte.
- Để biểu diễn một ký tự, ta có thể sử dụng một trong số các cách sau đây:
  - Đặt ký tự trong cặp dấu nháy đơn. Ví dụ 'A', '0'.

- Dùng hàm CHR(n) (trong đó n là mã ASCII của ký tự cần biểu diễn). Ví dụ CHR(65) biểu diễn ký tự 'A'.
- Dùng ký hiệu #n (trong đó n là mã ASCII của ký tự cần biểu diễn). Ví dụ #65.

- Các phép toán: =, >, >=, <, <=,<>.

*** Các hàm trên kiểu ký tự:**

- **UPCASE(ch)**: Trả về ký tự in hoa tương ứng với ký tự ch. Ví dụ: UPCASE('a') = 'A'.
- **ORD(ch)**: Trả về số thứ tự trong bảng mã ASCII của ký tự ch. Ví dụ ORD('A')=65.
- **CHR(n)**: Trả về ký tự tương ứng trong bảng mã ASCII có số thứ tự là n. Ví dụ: CHR(65)='A'.
- **PRED(ch)**: cho ký tự đứng trước ký tự ch. Ví dụ: PRED('B')='A'.
- **SUCC(ch)**: cho ký tự đứng sau ký tự ch. Ví dụ: SUCC('A')='B'.

II. KHAI BÁO HẰNG

- Hằng là một đại lượng có giá trị không thay đổi trong suốt chương trình.
- Cú pháp: **CONST<Tên hằng> = <Giá trị>;**

hoặc: **CONST<Tên hằng>: = <Biểu thức hằng>;**

Ví dụ:

```
CONST Max = 100;
      Name = 'Tran Van Hung';
      Continue = FALSE;
      Logic = ODD(5);    {Logic =TRUE}
```

Chú ý: *Chỉ các hàm chuẩn dưới đây mới được cho phép sử dụng trong một biểu thức hằng*:

ABS CHR HI LO LENGTH ODD ORD
PTR ROUND PRED SUCC SIZEOF SWAP TRUNC

III. KHAI BÁO BIẾN

- Biến là một đại lượng mà giá trị của nó có thể thay đổi trong quá trình thực hiện chương trình.
- Cú pháp: **VAR <Tờn biến>[,<Tờn biến 2>,...] : <Kiểu dữ liệu>;**

Ví dụ:

```
VAR   x, y: Real; {Khai báo hai biến x, y có kiểu là Real}
      a, b: Integer; {Khai báo hai biến a, b có kiểu integer}
```

Chú ý: *Ta có thể vừa khai báo biến, vừa gán giá trị khởi đầu cho biến bằng cách sử dụng cú pháp như sau*:

CONST<Tên biến>: <Kiểu> = <Giá trị>;

Ví dụ:

```
CONST x:integer = 5;
```

Với khai báo biến x như trên, trong chương trình giá trị của biến x có thể thay đổi. (Điều này không đúng nếu chúng ta khai báo x là hằng).

IV. ĐỊNH NGHĨA KIỂU

- Ngoài các kiểu dữ liệu do Turbo Pascal cung cấp, ta có thể định nghĩa các kiểu dữ liệu mới dựa trên các kiểu dữ liệu đã có.
- Cú pháp: **TYPE <Tên kiểu> = <Mô tả kiểu>;**
  VAR <Tên biến>:<Tên kiểu>;

Ví dụ:

```
TYPE  Sothuc = Real;
      Tuoi = 1..100;
      ThuNgay = (Hai,Ba,Tu, Nam, Sau, Bay, CN)
VAR   x :Sothuc;
      tt : Tuoi;
      Day: ThuNgay;
```

V. BIỂU THỨC

Biểu thức (expression) là công thức tính toán mà trong đó bao gồm các phép toán, các hằng, các biến, các hàm và các dấu ngoặc đơn.

Ví dụ: (x +sin(y))/(5-2*x) biểu thức số học

(x+4)*2 = (8+y) biểu thức logic

Trong một biểu thức, thứ tự ưu tiên của các phép toán được liệt kê theo thứ tự sau:

- Lời gọi hàm.
- Dấu ngoặc ()
- Phép toán một ngôi (NOT, -).
- Phép toán *, /, DIV, MOD, AND.
- Phép toán +, -, OR, XOR
- Phép toán so sánh =, <, >, <=, >=, <>, IN

VI. CÂU LỆNH

6.1. Câu lệnh đơn giản

- ***Câu lệnh gán (:=)***: <Tên biến>**:=**<Biểu thức>;
- Các lệnh xuất nhập dữ liệu: **READ/READLN, WRITE/WRITELN.**
- Lời gọi hàm, thủ tục.

6.2. Câu lệnh có cấu trúc

- Câu lệnh ghép: **BEGIN... END;**
- Các cấu trúc điều khiển: **IF.., FOR..., WHILE...,.....**

6.3. Các lệnh xuất nhập dữ liệu

6.3.1. Lệnh xuất dữ liệu

Để xuất dữ liệu ra màn hình, ta sử dụng ba dạng sau:

(1)WRITE(<tham số 1> [, <tham số 2>,...]);

(2)WRITELN(<tham số 1> [, <tham số 2>,...]);

(3)WRITELN;

Các thủ tục trên có chức năng như sau:

(1) Sau khi xuất giá trị của các tham số ra màn hình thì con trỏ không xuống dòng.

(2) Sau khi xuất giá trị của các tham số ra màn hình thì con trỏ xuống đầu dòng tiếp theo.

(3) Xuất ra màn hình một dòng trống.

Các tham số có thể là các hằng, biến, biểu thức. Nếu có nhiều tham số trong câu lệnh thì các tham số phải được phân cách nhau bởi dấu phẩy.

Khi sử dụng lệnh WRITE/WRITELN, ta có hai cách viết: ***không qui cách*** và ***có qui cách***:

- ***Viết không qui cách***: dữ liệu xuất ra sẽ được canh lề ở phía bên trái. Nếu dữ liệu là số thực thì sẽ được in ra dưới dạng biểu diễn khoa học.

Ví dụ: WRITELN(x); WRITE(sin(3*x));

- ***Viết có qui cách***: dữ liệu xuất ra sẽ được canh lề ở phía bên phải.

Ví dụ: WRITELN(x:5); WRITE(sin(13*x):5:2);

Câu lệnh	Kết quả trên màn hình
Writeln('Hello');	Hello
Writeln('Hello':10);	Hello
Writeln(500);	500
Writeln(500:5);	500
Writeln(123.457)	1.2345700000E+02
Writeln(123.45:8:2)	123.46

6.3.2. Nhập dữ liệu

Để nhập dữ liệu từ bàn phím vào các biến có kiểu dữ liệu chuẩn (trừ các biến kiểu BOOLEAN), ta sử dụng cú pháp sau đây:

READLN(<biến 1> [,<biến 2>,...,<biến n>]);

Chú ý: *Khi gặp câu lệnh* **READLN;** *(không có tham số), chương trình sẽ dừng lại chờ người sử dụng nhấn phím ENTER mới chạy tiếp.*

6.4. Các hàm và thủ tục thường dùng trong nhập xuất dữ liệu

- Hàm **KEYPRESSED**: Hàm trả về giá trị TRUE nếu như có một phím bất kỳ được nhấn, nếu không hàm cho giá trị là FALSE.
- Hàm **READKEY**: Hàm có chức năng đọc một ký tự từ bộ đệm bàn phím.
- Thủ tục **CLRSCR**: Xoá màn hình và đưa con trỏ về góc trên bên trái màn hình.

BÀI TẬP MẪU

Bài tập 2.1: Viết chương trình nhập vào độ dài hai cạnh của tam giác và góc giữa hai cạnh đó, sau đó tính và in ra màn hình diện tích của tam giác.

Ý tưởng: Công thức tính diện tích tam giác: S = $\frac{1}{2}a.b.\sin(\theta)$ với a,b là độ dài 2 cạnh và θ là góc kẹp giữa 2 cạnh a và b.

```
Program Tinh_dien_tich_tam_giac;
Var a,b,goc,dientich: Real;
Begin
    Write('Nhap vao do dai canh thu nhat: '); Readln(a);
    Write('Nhap vao do dai canh thu hai: '); Readln(b);
  Write('Nhap vao goc giua hai canh: '); Readln(goc);
    Dientich:=a*b*sin(goc)/2;
    Writeln('Dien tich cua tam giac la: ',Dientich:0:2);
  Readln;
End.
```

Bài tập 2.2: Viết chương trình tính $\sqrt[n]{x}$, x>0.

Ý tưởng: Ta có: $\sqrt[n]{x} = x^{\frac{1}{n}} = e^{\frac{1}{n}\ln x}$

```
Program Tinh_can_bac_n_cua_x;
Var x,S: Real;
    n: Word;
Begin
  Write('Nhap vao n= '); Readln(n);
  Write('Nhap vao x= '); Readln(x);
  S:=EXP(1/n*LN(x));
  Writeln('S = ',S:0:2);
  Readln;
End.
```

Bài tập 2.3: Viết chương trình nhập vào 2 số a, b. Sau đó hoán đổi giá trị của 2 số đó:

a/ Cho phép dùng biến trung gian.

```
Program Swap;
Var a,b,tam: Integer;
Begin
  Write('Nhap vao a= '); Readln(a);
  Write('Nhap vao b= '); Readln(b);
  tam:=a;   {tam lấy giá trị của a}
  a:=b;     {a lấy giá trị của b}
  b:=tam; {b lấy lại giá trị của tam}
  Writeln('a = ',a,' b = ',b);
  Readln;
End.
```

b/ Không được phép dùng biến trung gian.

```
Program Swap;
Var a,b: Integer;
Begin
  Write('Nhap vao a= '); Readln(a);
  Write('Nhap vao b= '); Readln(b);
  a:=a+b;   {a lấy tổng giá trị của a+b}
  b:=a-b;   {b lấy giá trị của a}
  a:=a-b;   {a lấy lại giá trị của b}
  Writeln('a = ',a,' b = ',b);
  Readln;
End.
```

BÀI TẬP TỰ GIẢI

Bài tập 2.4: Viết chương trình nhập vào các số nguyên: a, b, x, y,... sau đó in ra màn hình kết quả của các biểu thức sau:

a/ $\dfrac{x+y}{2+\dfrac{x}{y}}$ b/ $\dfrac{(a+4)(b-2c+3)}{\dfrac{r}{2h}-9(a-1)}$ c/ x^y, x>0 d/ $e^{\sqrt{|a+\sin^2(x)-x|}}$

Bài tập 2.5: Viết chương trình tính siện tích tam giác theo công thức sau:

S = $\sqrt{p(p-a)(p-b)(p-c)}$ với p = $\dfrac{1}{2}$(a+b+c)

Bài tập 2.6: Viết chương trình tính khoảng cách từ một điểm $I(x_i,y_i)$ đến đường thẳng có phương trình D: Ax + By + C = 0.

Gợi ý: Công thức tính khoảng cách: h = $\dfrac{A.x_i + B.y_i + C}{\sqrt{A^2+B^2}}$

CHƯƠNG 3: CÁC CÂU LỆNH CÓ CẤU TRÚC

I. CÂU LỆNH RẼ NHÁNH

1.1. Lệnh IF

Cú pháp: ❶ **Dạng thiếu IF *<điều kiện>* THEN *<câu lệnh>;***

❷ **Dạng đủ IF *<điều kiện>* THEN *<câu lệnh 1>* ELSE *<câu lệnh 2>*;**

Chú ý: *Khi sử dụng câu lệnh IF thì đứng trước từ khoá ELSE không được có dấu chấm phẩy (;).*

Trong đó: <điều kiện> là phép toán qua hệ hay biểu thức logic.

<câu lệnh> hoặc ***<câu lệnh 1>, <câu lệnh 2>*** là câu lênh đơn hoặc câu lệnh ghép trong pascal.

II. CÂU LỆNH LẶP

2.1. Vòng lặp xác định Có hai dạng sau:

❶ ***Dạng tiến:*** **FOR <biến đếm>:=<giá trị đầu> TO <giá trị cuối> DO *<câu lệnh>*;**

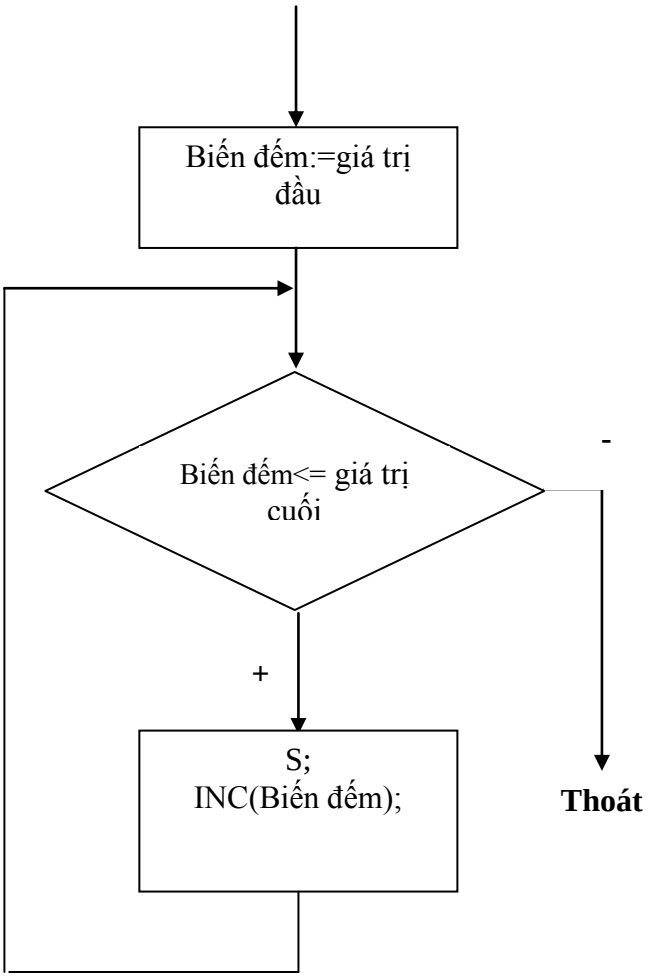
❷ ***Dạng lùi:*** **FOR <biến đếm>:=<giá trị cuối> DOWNTO <giá trị đầu> DO *<câu lệnh>*;**

Trong đó: - *<biến đếm>* là biến đơn, kiểu nguyên, **<giá trị đầu>, <giá trị cuối>** cùng kiểu với kiểu biến đếm, vòng lặp thực hiện khi **<giá trị đầu> <= <giá trị cuối>.**

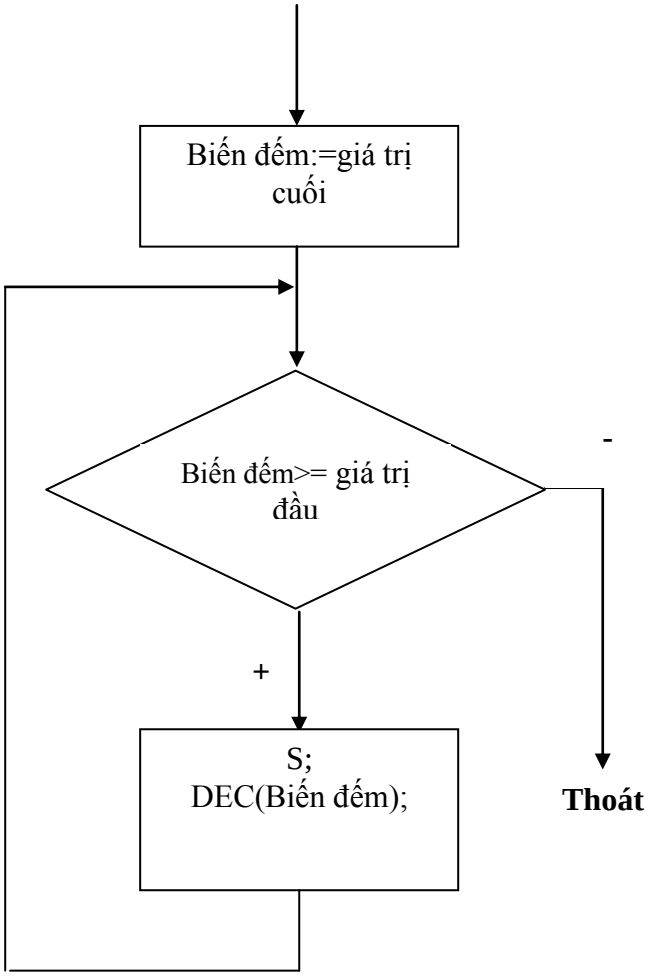
- ***<câu lệnh>*** là câu lệnh đơn hay câu lênh ghép trong pascal.

Sơ đồ thực hiện vòng lặp FOR:

Dạng tiến



Dạng lùi



Chú ý: Khi sử dụng câu lệnh lặp FOR cần chú ý các điểm sau:

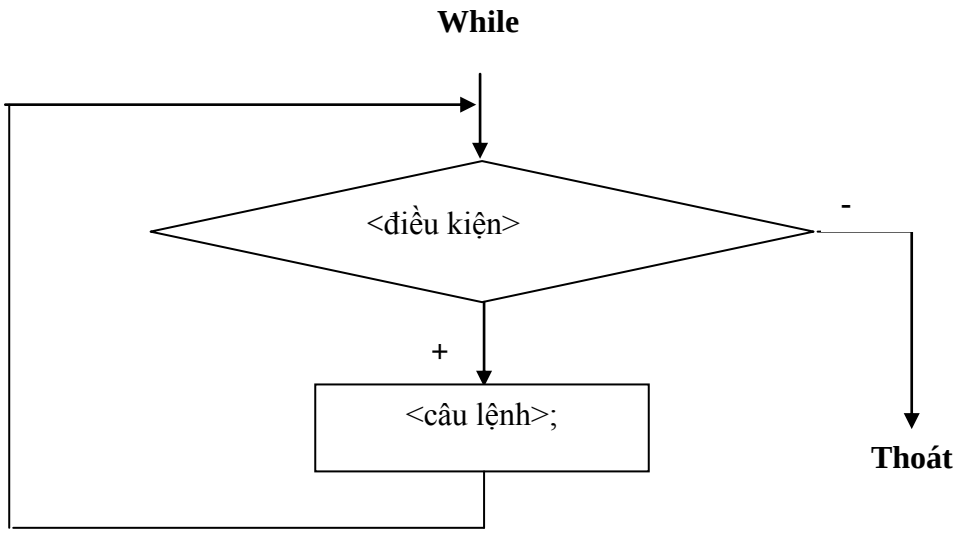
- *Không nên tuỳ tiện thay đổi giá trị của biến đếm bên trong vòng lặp FOR vì làm như vậy có thể sẽ không kiểm soát được biến đếm.*
- *<Giá trị đầu> và <Giá trị cuối> trong câu lệnh FOR sẽ được xác định ngay khi vào đầu vòng lặp. Do đó cho dù trong vòng lặp ta có thay đổi giá trị của nó thì số lần lặp cũng không thay đổi.*
- *Sau mỗi lần lặp bản thân biến đếm tự động tăng lên 1 đơn vị.*

5.3.2. Vòng lặp không xác định

Cấu trúc:* WHILE *<điều kiện>* do *<câu lệnh>;

Ý nghĩa: Dạng WHILE trong khi **<điều kiện>** thì tiếp tục thực hiện công việc **<câu lệnh>;**

- ***<điều kiện>*** là phép toán quan hệ hay biểu thức logic.
- ***<câu lệnh>*** là câu lênh đơn hoặc là câu lệnh ghép.



Một số ví dụ minh họa.

Bài tập 3.1: Viết chương trình nhập vào một số nguyên và kiểm tra xem số vừa nhập là số chẵn hay số lẻ.

```
Uses crt;
Var x:integer;
Begin
 Write('Nhap vao mot so nguyen : '); Readln(x);
  If x MOD 2=0 Then   Writeln('So vua nhap vao la so chan')
  Else  Writeln('So vua nhap vao la so le');
  Readln;
End.
```

Bài tập 3.2: Viết chương trình giải phương trình bậc nhất ax+b=0

```
Uses Crt;
Var a,b,x : real;
Begin
  Write('a = '); Readln(a);
  Write('b = '); Readln(b);
  If a = 0 Then   { Nếu a bằng 0 }
    If b = 0 Then { Trường hợp a = 0 và b = 0 }
      Writeln('Phuong trinh co vo so nghiem')
    Else { Trường hợp a=0 và b ≠ 0 }
      Writeln('Phuong trinh vo nghiem')
      Else { Trường hợp a ≠ 0 }
      Begin
        x:= -b/a;
        Writeln('Phuong trinh co nghiem la :',x:0:2);
      End;
  Readln;
End.
```

Bài tập rèn luyện:

Bài tập 3.3: Viết chương trình nhập vào tuổi của một người và cho biết người đó là thiếu niên, thanh niên, trung niên hay lão niên. Biết rằng: nếu tuổi nhỏ hơn 18 là thiếu niên, từ 18 đến 39 là thanh niên, từ 40 đến 60 là trung niên và lớn hơn 60 là lão niên.

Bài tập 3.4: Viết chương trình tính tổng S = 1+2+...+N

Bài tập 3.5: Viết chương trình nhập vào N số nguyên từ bàn phím. Hãy tính và in ra màn hình tổng của các số vừa được nhập vào.

Ý tưởng: Dùng phương pháp cộng dồn. Cho vòng lặp FOR chạy từ 1 tới N, ứng với lần lặp thứ i, ta nhập vào số nguyên X và đồng thời cộng dồn X vào biến S.

Bài tập 3.6: Viết chương trình nhập vào các số nguyên cho đến khi nào gặp số 0 thì kết thúc. Hãy đếm xem có bao nhiêu số chẵn vừa được nhập vào.

Ý tưởng: Bài toán này không biết chính xác số lần lặp nên ta không thể dùng vòng lặp FOR. Vì phải nhập vào số nguyên N trước, sau đó mới kiểm tra xem N=0? Do đó ta nên dùng vòng lặp REPEAT.

Bài tập 3.7: Viết chương trình tính số Pi với độ chính xác Epsilon, biết: Pi/4 = 1-1/3+1/5-1/7+...

Ý tưởng: Ta thấy rằng, mẫu số là các số lẻ có qui luật: 2*i+1 với i=1,...,n. Do đó ta dùng i làm biến chạy.

Vì tính số Pi với độ chính xác **Epsilon** nên không biết trước được cụ thể số lần lặp, do đó ta phải dùng vòng lặp WHILE. Có nghĩa là phải lặp cho tới khi t=4/(2*i+1) ≤ Epsilon thì dừng.

Bài tập 3.8: Viết chương trình nhập vào số nguyên N. In ra màn hình tất cả các ước số của N.

Ý tưởng: Cho biến i chạy từ 1 tới N. Nếu N MOD i=0 thì viết i ra màn hình.

Bài tập 3.9: Viết chương trình tìm USCLN và BSCNN của 2 số a, b được nhập vào từ bàn phím.

Ý tưởng:- Tìm USCLN: Lấy số lớn trừ số nhỏ cho đến khi a=b thì dừng. Lúc đó: USCLN=a.

- BSCNN(a,b) = a*b DIV USCLN(a,b).

Bài tập 3.10: Viết chương trình tìm các số có 3 chữ số $\overline{abc}$ sao cho: $\overline{abc} = a^3 + b^3 + c^3$.

Ý tưởng: Dùng phương pháp vét cạn. Ta biết rằng: a có thể có giá trị từ 1→9 (vì a là số hàng trăm), b,c có thể có giá trị từ 0→9. Ta sẽ dùng 3 vòng lặp FOR lồng nhau để duyệt qua tất cả các trường hợp của a,b,c.

Ứng với mỗi bộ abc, ta sẽ kiểm tra: Nếu $100.a + 10.b + c = a^3 + b^3 + c^3$ thì in ra bộ abc đó.

Bài tập 3.11: Viết chương trình nhập vào số tự nhiên N rồi thông báo lên màn hình số đó có phải là số nguyên tố hay không.

Ý tưởng: N là số nguyên tố nếu N không có ước số nào từ 2 → N div 2. Từ định nghĩa này ta đưa ra giải thuật:

- Đếm số ước số của N từ 2 → N div 2 lưu vào biến d.
- Nếu d=0 thì N là số nguyên tố.

Bài tập 3.12: Viết chương trình giải phương trình bậc hai: $ax^2 + bx + c = 0$, a≠0.

Gợi ý: - Tính Delta=b*b-4*a*c.

- Biện luận:

  Delta<0: Phương trình vô nghiệm.

  Delta=0: Phương trình có nghiệm kép: x = -b/(2*a).

  Delta>0: Phương trình có 2 nghiệm phân biệt: $x_{1,2}$ = (-b±SQRT(Delta))/(2*a).

Bài tập 3.13: Viết chương trình nhập vào từ bàn phím: giờ, phút, giây. Cọng thêm một số giây cũng được nhập từ bàn phím. Hãy in ra kết quả sau khi cọng xong.

Gợi ý: - Gọi số giây được cộng thêm là: ss. Gán giây:=giây+ss.

- Nếu giây≥60 thì: phút:=phút + giây DIV 60 và giây:=giây MOD 60.
- Nếu phút≥60 thì: giờ:=giờ + phút DIV 60 và phút:=phút MOD 60.

Bài tập 3.14: Viết chương trình tìm Max, Min của 4 số: a, b, c, d.

Bài tập 3.15: Viết chương trình in ra màn hình các giá trị của bảng mã ASCII từ 0→255.

Gợi ý: Cho biến i chạy từ 0 → 255. In ra màn hình i và CHR(i).

Bài tập 3.16: Viết chương trình in ra màn hình các số nguyên từ 1 đến 100 sao cho cứ 10 số thì xuống dòng.

Gợi ý: Cho biến i chạy từ 1 → 100. In ra màn hình i và kiểm tra: nếu i MOD 10=0 thì WRITELN.

Bài tập 3.17: Viết chương trình in ra màn hình bảng cữu chương.

Gợi ý: Dùng 2 vòng lặp FOR lồng nhau: i là số bảng cữu chương (2...9), j là số thứ tự trong từng bảng cữu chương (1...10).

```
For i:=2 To 9 Do
   For j:=1 To 10 Do Writeln(i,'x',j,'=',i*j);
```

Bài tập 3.18: Viết chương trình tính các tổng sau:

S0 = n! = 1*2*...*n {***n giai thừa***}

S1 = 1 + 1/2 +... + 1/n

S2 = 1 + 1/2! +... + 1/n!

S3 = 1 + x + x2/2! + x3/3! +... + xn/n!

S4 = 1 - x + x2/2! - x3/3! +... + $(-1)^n x^n/n!$

S5 = 1 + sin(x) + $\sin^2(x)$ +... + $\sin^n(x)$.

Bài tập 3.19: Viết chương trình nhập vào một số nguyên dương. Hãy thông báo lên màn hình số đó có bao nhiêu chữ số và tổng các chữ số của số đó.

Gợi ý: Dùng vòng lặp WHILE. Trong khi N>0 thì: lấy ra chữ số cuối cùng của N để tính bằng phép toán MOD 10, sau đó bỏ bớt đi chữ số cuối cùng của N bằng phép toán DIV 10.

Bài tập 3.20: Viết chương trình in ra màn hình tất cả các số nguyên tố từ 2 đến N. Với N được nhập từ bàn phím.

Bài tập 3.21: Viết chương trình phân tích một số ra thừa số nguyên tố. Ví dụ: N=100 sẽ in ra màn hình:

```
100 | 2
 50 | 2
 25 | 5
  5 | 5
  1 |
```

CHƯƠNG 4: CHƯƠNG TRÌNH CON - THỦ TỤC VÀ HÀM

I. KHÁI NIỆM VỀ CHƯƠNG TRÌNH CON

Chương trình con (CTC) là một đoạn chương trình thực hiện trọn vẹn hay một chức năng nào đó. Trong Turbo Pascal, có 2 dạng CTC:

- Thủ tục (PROCEDURE): Dùng để thực hiện một hay nhiều nhiệm vụ nào đó.
- Hàm (FUNCTION): Trả về một giá trị nào đó (có kiểu vô hướng, kiểu string hoặc kiểu con trỏ). Hàm có thể sử dụng trong các biểu thức.

Ngoài ra, trong Pascal còn cho phép các CTC lồng vào nhau.

II. CẤU TRÚC CHUNG CỦA MỘT CHƯƠNG TRÌNH CÓ SỬ DỤNG CTC

```
PROGRAM Tên_chương_trình;
USES CRT;
CONST.....;
TYPE.....;
VAR.....;
PROCEDURE THUTUC[(Các tham số)];
[Khai báo Const, Type, Var]
BEGIN
   .....
END;
FUNCTION HAM[(Các tham số)]:<Kiểu dữ liệu>;
[Khai báo Const, Type, Var]
BEGIN
   .....
   HAM:=<Giá trị>;
END;

BEGIN {Chương trình chính}
   .....
   THUTUC[(...)];
   .....
   A:= HAM[(...)];
   .....
END.
```

Chú ý: Trong quá trình xây dựng CTC, khi nào thì nên dùng thủ tục/hàm?

Dùng hàm	Dùng thủ tục
- Kết quả của bài toán trả về ***1 giá trị duy nhất (kiểu vô hướng, kiểu string hoặc kiểu con trỏ).*** - Lời gọi CTC cần nằm trong các biểu thức tính toán.	- Kết quả của bài toán ***không trả về giá trị nào*** hoặc trả về ***nhiều giá trị*** hoặc ***trả về kiểu dữ liệu có cấu trúc (Array, Record, File).*** - Lời gọi CTC không nằm trong các biểu thức tính toán.

Ví dụ 1: Viết CTC để tính n! = 1.2...n.

Ý tưởng: Vì bài toán này trả về 1 giá trị duy nhất nên ta dùng hàm.

```
Function GiaiThua(n:Word):Word;
Var P, i:Word;
Begin
  P:=1;
  For i:=1 To n Do P:=P*i;
```

```
    GiaiThua:=P;
  End;
```

Ví dụ 2: Viết chương trình con để tìm điểm đối xứng của điểm (x,y) qua gốc tọa độ.

Ý tưởng: Vì bài toán này trả về tọa độ điểm đối xứng (xx,yy) gồm 2 giá trị nên ta dùng thủ tục.

```
Procedure DoiXung(x,y:Integer; Var xx,yy:Integer);
Begin
  xx:=-x;
  yy:=-y;
End;
```

CHÚ Ý: Trong 2 ví dụ trên:

- **n, x, y** được gọi là **tham trị** (*không có từ khóa* **var** *đứng trước*) vì sau khi ra khỏi CTC giá trị của nó ***không bị thay đổi***.
- **xx, yy** được gọi là **tham biến** (*có từ khóa* **var** *đứng trước*) vì sau khi ra khỏi CTC giá trị của nó ***bị thay đổi***.

III. BIẾN TOÀN CỤC VÀ BIẾN ĐỊA PHƯƠNG

- **Biến toàn cục**: là các biến được khai báo trong chương trình chính. Các biến này có tác dụng ở mọi nơi trong toàn bộ chương trình.
- **Biến địa phương**: là các biến được khai báo trong các CTC. Các biến này chỉ có tác dụng trong phạm vi CTC đó mà thôi.

Chú ý: Trong một CTC, nếu biến toàn cục trùng tên với biến địa phương thì biến địa phương được ưu tiên hơn.

Ví dụ:

```
Program KhaoSatBien;
Var a,b: Integer; {biến toàn cục}
Procedure ThuBien;
Var a: Integer; {biến địa phương}
Begin
  a:=10;
  Writeln('A=',a,'B=',b);
End;
Begin
  a:=50;
  b:=200;
  ThuBien;              {A=10 B=200}
  Writeln('A=',a,'B=',b);   {A=50 B=200}
End.
```

Một số ví dụ minh họa

Ví dụ 4.1: Viết hàm tìm Max của 2 số thực x,y.

```
Var a,b:Real;
Function Max(x,y:Real):Real;
Begin
  If x>y Then Max:=x Else Max:=y;
End;
Begin
  Write('Nhap a='); Readln(a);
  Write('Nhap b='); Readln(b);
  Writeln('So lon nhat trong 2 so la: ', Max(a,b));
```

```
    Readln;
  End.
```

Ví dụ 4.2: Viết thủ tục để hoán đổi hai gía trị x,y cho nhau.

```
Var a,b:Real;
Function Swap(Var x,y:Real);
Var Tam:Real;
Begin
  Tam:=x; x:=y; y:=Tam;
End;
Begin
  Write('Nhap a='); Readln(a);
  Write('Nhap b='); Readln(b);
  Swap(a,b);
  Writeln('Cac so sau khi hoan doi: a=', a:0:2,' b=',b:0:2);
  Readln;
End.
```

Ví dụ 4.3: Viết hàm XMU(x:Real;n:Byte):Real; để tính giá trị x^n.

```
  Var x:Real;
      n:Byte;
  Function XMU(x:Real;n:Byte):Real;
  Var i:Byte; S:Rea;
  Begin
    S:=1;
    For i:=1 To n Do S:=S*x;
    XMU:=S;
  End;
  Begin
    Write('Nhap x='); Readln(x);
    Write('Nhap n='); Readln(n);
    Writeln('x mu n = ', XMU(x,n):0:2);
    Readln;
  End.
```

Ví dụ 4.4: Viết thủ tục PHANTICH(n:Integer); để phân tích số nguyên n ra thừa số nguyên tố.

```
  Uses crt;
  Var n:Integer;
  Procedure PHANTICH(n:Integer);
  Var i:Integer;
  Begin
   i:=2;
   While n<>1 Do
   Begin
   While n MOD i=0 Do
   Begin
   Writeln(n:5,'|',i:2);
   n:=n Div i;
   End;
   i:=i+1;
```

```
   End;
   Writeln(n:5,'|');
  End;
  Begin
   Write('Nhap n='); Readln(n);
   PHANTICH(n);
   Readln;
  End.
```

BÀI TẬP

Bài tập 4.1: Viết 2 hàm tìm Max, min của 3 số thực.

Bài tập 4.2: Viết các hàm và thủ tục để tính:

$S_1 = 1+2+3+.....+n$;

$S_2 = 1+1/2+.....+1/n$;

$S_3 = 1-1/2+.....+(-1)^{n+1}\ 1/n$

$S_4 = 1 + \sin(x) + \sin^2(x) +.....+ \sin^n(x)$

Bài tập 4.3: Viết hàm đệ quy để tính C^k_n biết :

$C^n_n = 1$, $C^0_n = 1$, $C^k_n = C^{k-1}_{n-1} + C^k_{n-1}$.

Bài tập 4.4: Lập hàm để tính dãy Fibonaci:

$$F(n) = \begin{cases} 1 \ , & n = 1 \vee n = 2 \\ F(n-1) + F(n-2) \ , & n > 2 \end{cases}$$

Bài tập 4.5: Viết hàm tìm USCLN của 2 số.

Bài tập 4.6: Viết thủ tục để in ra màn hình số đảo ngược của một số nguyên cho trước.

Bài tập 4.7: Xây dựng một chương trình chứa các thủ tục và hàm thực hiện các chức năng sau:

- Giải phương trình bặc nhất.
- Giải phương trình bặc hai.
- Tìm Max/Min của 2 số a,b.
- Tìm USCLN và BSCNN của 2 số nguyên a,b.
- Kiểm tra số nguyên dương n có phải là số nguyên tố hay không?
- Kiểm tra số nguyên dương n có phải là số hoàn thiện hay không?
- Đổi một số nguyên dương n sang dạng nhị phân.
- In ra màn hình bảng cữu chương từ $2 \rightarrow 9$.

CHƯƠNG 5: DỮ LIỆU KIỂU MẢNG (ARRAY – DÃY SỐ)

I. KHAI BÁO MẢNG

Cú pháp:

TYPE <Kiểu mảng> = ARRAY [chỉ số] OF <Kiểu dữ liệu>;

VAR <Biến mảng>:<Kiểu mảng>;

hoặc khai báo trực tiếp:

VAR <Biến mảng> : ARRAY [chỉ số] OF <Kiểu dữ liệu>;

Ví dụ:

```
TYPE  Mangnguyen = Array[1..100] of Integer;
      Matrix = Array[1..10,1..10] of Integer;
      MangKytu = Array[Byte] of Char;
VAR   A: Mangnguyen;
      M: Matrix;
      C: MangKytu;
```

hoặc:

```
VAR   A: Array[1..100] of Integer;
      C: Array[Byte] of Char;
```

II. XUẤT NHẬP TRÊN DỮ LIỆU KIỂU MẢNG

- Để truy cập đến phần tử thứ k trong mảng một chiều A, ta sử dụng cú pháp: A[k].
- Để truy cập đến phần tử (i,j) trong mảng hai chiều M, ta sử dụng cú pháp: M[i,j].
- Có thể sử dụng các thủ tục READ(LN)/WRITE(LN) đối với các phần tử của biến kiểu mảng.

Lưu ý: Một số kỹ thuật xử lý trên dãy số

- Thuật toán tìm Max, Min,
- Thuật toán sắp xếp,
- Thuật toán tìm kiếm nhị phân,
- Thuật toán xử lý mảng vòng, kỹ thuật đánh dấu....

Mộ số ví dụ minh họa

Ví dụ 5.1: Viết chương trình tìm giá trị lớn nhất của một mảng chứa các số nguyên gồm N phần tử.

Ý tưởng: - Cho số lớn nhất là số đầu tiên: Max:=a[1].

- Duyệt qua các phần tử a[i], với i chạy từ 2 tới N: Nếu a[i]>Max thì thay Max:=a[i];

```
Uses Crt;
Type Mang = ARRAY[1..50] Of Integer;
Var A:Mang;
    N,i,Max:Integer;
Begin
  {Nhập mảng}
  Write('Nhap N='); Readln(N);
  For i:=1 To N Do
    Begin
      Write('A[',i,']='); Readln(A[i]);
    End;
  {Tìm phần tử lớn nhất}
  Max:=A[1];
```

```
    For i:=2 To N Do
      If Max<A[i] Then Max:=A[i];
    {In kết quả ra màn hình}
    Writeln('Phan tu lon nhat cua mang: ', Max);
    Readln;
  End.
```

Ví dụ 5.2: Viết chương trình tính tổng bình phương của các số âm trong một mảng gồm N phần tử.

Ý tưởng: Duyệt qua tất cả các phần tử A[i] trong mảng: Nếu A[i]<0 thì cộng dồn $(A[i])^2$ vào biến S.

```
Uses Crt;
Type Mang = ARRAY[1..50] Of Integer;
Var A:Mang;
    N,i,S:Integer;
Begin
  {Nhập mảng}
  Write('Nhap N='); Readln(N);
  For i:=1 To N Do
    Begin
      Write('A[',i,']='); Readln(A[i]);
    End;
  {Tính tổng}
  S:=0;
  For i:=1 To N Do
    If A[i]<0 Then S:=S+A[i]*A[i];
  {In kết quả ra màn hình}
  Writeln('S= ', S);
  Readln;
End.
```

Ví dụ 5.3: Viết chương trình nhập vào một mảng gồm N số nguyên. Sắp xếp lại mảng theo thứ tự tăng dần và in kết quả ra màn hình.

Ý tưởng: Cho biến i chạy từ 1 đến N-1, đồng thời cho biến j chạy từ i+1 đến N: Nếu A[i]>A[j] thì đổi chỗ A[i], A[j].

```
  Uses Crt;
  Type Mang = ARRAY[1..50] Of Integer;
  Var A:Mang;
      N,i,j,Tam:Integer;
  Begin
    {Nhập mảng}
    Write('Nhap N='); Readln(N);
    For i:=1 To N Do
      Begin
        Write('A[',i,']='); Readln(A[i]);
      End;
    {Sắp xếp}
    For i:=1 To N-1 Do
```

```
        For j:=i+1 To N Do
          If A[i]>A[j] Then
            Begin
              Tam:=A[i]; A[i]:=A[j]; A[j]:=Tam;
            End;
      {In kết quả ra màn hình}
      Writeln('Ket qua sau khi sap xep:');
      For i:=1 To N Do Write(A[i]:5);
      Readln;
    End.
```

Ví dụ 5.4: Viết chương trình nhập vào một mảng A gồm N số nguyên và nhập thêm vào một số nguyên X. Hãy kiểm tra xem phần tử X có trong mảng A hay không?

Ý tưởng: Dùng thuật toán tìm kiếm tuần tự. So sánh x với từng phần tử của mảng A. Thuật toán dừng lại khi x=A[i] hoặc i>N.

Nếu x=A[i] thì vị trí cần tìm là i, ngược lại thì kết quả tìm là 0 (không tìm thấy).

```
Uses Crt;
Type Mang = ARRAY[1..50] Of Integer;
Var A:Mang;
    N,i,x:Integer;
Function TimKiem(x, N: Integer; A:Mang):Integer;
Var i:Integer;
  Begin
    I:=1;
    While (I <= N) and (X<>A[I]) do I:=I+1;
    If I <= N Then Timkiem:=I Else Timkiem:=0;
  End;
Begin
  {Nhập mảng}
  Write('Nhap N='); Readln(N);
  For i:=1 To N Do
    Begin
      Write('A[',i,']='); Readln(A[i]);
    End;
  Write('Nhap X='); Readln(x);
  {Kết quả tìm kiếm}
  If TimKiem(X,N,A)<>0 Then
    Writeln('Vi tri cua X trong mang la:', TimKiem(X,N,A))
  Else Writeln('X khong co trong mang.');
  Readln;
End.
```

Ví dụ 5.5: Giả sử mảng A đã được sắp xếp theo thứ tự tăng dần. Viết hàm để kiểm tra xem phần tử X có trong mảng A hay không?

Ý tưởng: So sánh x với phần tử ở giữa mảng A[giua]. Nếu x=A[giua] thì dừng (vị trí cần tìm là chỉ số của phần tử giữa của mảng). Ngược lại, nếu x>A[giua] thì tìm ở đoạn sau của mảng [giua+1,cuoi], ngược lại thì tìm ở đoạn đầu của mảng [dau,giua-1].

Sau đây là hàm cài đặt cho thuật toán này:

```
Function TimKiemNhiPhan(X, N: Integer; A: Mang):Integer;
Var dau,cuoi,giua:Integer;
    Found:Boolean;
Begin
  dau:=1; {điểm mút trái của khoảng tìm kiếm}
  cuoi:=N; {điểm mút phải của khoảng tìm kiếm}
  Found:=False; {chưa tìm thấy}
  While (dau <=cuoi) and (Not Found) Do
    Begin
      giua:=(dau + cuoi) Div 2;
      If X = A[giua] Then Found:=True {đã tìm thấy}
      Else
        If X > A[giua] Then dau:=giua+1
        Else cuoi:=giua-1;
    End;
  If Found Then TimKiemNhiPhan:= giua Else TimKiemNhiPhan:=0;
End;
```

Ví dụ 5.6: Viết chương trình nhập vào 2 mảng số nguyên A, B đại diện cho 2 tập hợp (không thể có 2 phần tử trùng nhau trong một tập hợp). Trong quá trình nhập, phải kiểm tra: nếu phần tử vừa nhập vào đã có trong mảng thì không bổ sung vào mảng. In ra màn hình các phần tử là giao của 2 tập hợp A, B.

Ý tưởng: Duyệt qua tất cả các phần tử $a_i \in A$. Nếu $a_i \in B$ thì viết a_i ra màn hình.

```
Uses Crt;
Type Mang=ARRAY[1..50] Of Integer;
Var A,B:Mang;
 n,m:Byte;
Function KiemTra(x:Integer; n:Byte; A:Mang):Boolean;
Var i:Byte; Found:Boolean;
Begin
 Found:=False;
 i:=1;
 While (i<=n) AND (not Found) Do
 If x=A[i] Then Found:=True Else i:=i+1;
 KiemTra:=Found;
End;
Procedure NhapMang(Var n:Byte; Var A:Mang);
Var ch:Char;
 x:Integer;
Begin
 n:=0;
 Repeat
 Write('x='); Readln(x);
 If not KiemTra(x,n,A) Then
 Begin
 n:=n+1; A[n]:=x;
 End;
 Writeln('An ESC de ket thuc nhap!');
 ch:=Readkey;
 Until ch=#27;
End;
```

```
Procedure GiaoAB(n:Byte; A:Mang;m:Byte; B:Mang);
Var i:Byte;
Begin
 For i:=1 To n Do
 If KiemTra(A[i],m,B) Then Write(A[i]:4);
End;
Begin
 Clrscr;
 Writeln('Nhap mang A: ');
 NhapMang(n,A);
 Writeln('Nhap mang B: ');
 NhapMang(m,B);
 Writeln('Giao cua 2 mang A&B la: ');
 GiaoAB(n,A,m,B);
 Readln;
End.
```

Ví dụ 5.7: Cho một mảng số nguyên gồm n phần tử. Tìm dãy con gồm m phần tử (m≤n) sao cho dãy con này có tổng lớn nhất. (Dãy con là dãy các phần tử liên tiếp nhau trong mảng).

```
Uses Crt;
Type Mang=ARRAY[1..50] Of Integer;
Var A:Mang;
 n,m,i,j,k:Byte;
 S,Max:Integer;
Begin
 Write('So phan tu cua mang: n= '); Readln(n);
 For i:=1 To n Do
 Begin
 Write('a[',i,']='); Readln(a[i]);
 End;
 Write('Nhap so phan tu cua day con: m= '); Readln(m);
 k:=1; {Vị trí phần tử đầu tiên của dãy con}
 {Giả sử m phần tử đầu tiên của mảng A là dãy con có tổng lớn
nhất}
 Max:=0;
 For i:=1 To m Do Max:=Max+A[i];
{Tìm các dãy con khác}
 For i:=2 To n-m+1 Do
 Begin
 {Tính tổng của dãy con thứ i}
 S:=0;
 For j:=i To i+m-1 Do S:=S+A[j];
 If S>Max Then {Nếu dãy con tìm được có tổng lớn hơn dãy con
trước}
 Begin
 Max:=S; {Thay tổng mới}
 k:=i; {Thay vị trí đầu tiên của dãy con mới}
 End;
 End;
 Writeln('Day con co tong lon nhat la:');
 For i:=k To k+m-1 Do Write(A[i]:5);
 Readln;
End.
```

BÀI TẬP

Bài tập 5.1: Nhập vào một mảng các số nguyên.

a/ Xếp lại mảng đó theo thứ tự giảm dần.

b/ Nhập vào một số nguyên từ bàn phím. Chèn số đó vào mảng sao cho mảng vẫn có thứ tự giảm dần. (không được xếp lại mảng)

Gợi ý: - Tìm vị trí cần chèn: i.

- Đẩy các phần tử từ vị trí i tới n sang phải 1 vị trí.
- Gán: A[i]=x;

Bài tập 5.2: Cho 2 mảng số nguyên: Mảng A có m phần tử, mảng B có n phần tử.

a/ Sắp xếp lại các mảng đó theo thứ tự giảm dần.

b/ Trộn 2 mảng đó lại thành mảng C sao cho mảng C vẫn có thứ tự giảm dần (Không được xếp lại mảng C).

Gợi ý: - Dùng 2 chỉ số i,j để duyệt qua các phần tử của 2 mảng A, B và k là chỉ số cho mảng C.

- Trong khi (i<=m) và (j<=n) thì:

{Tức là khi đồng thời cả 2 dãy A, B đều chưa duyệt hết}

+ Nếu A[i]>B[j] thì: C[k]:=A[i]; i:=i+1;

+ Ngược lại: C[k]:=B[j]; j:=j+1;

- Nếu dãy nào hết trước thì đem phần còn lại của dãy kia bổ sung vào cuối dãy C.

Bài tập 5.3: Viết chương trình nhập vào một dãy số nguyên a_1, a_2,..., a_n. Tìm trong dãy {a} một dãy con tăng dần dài nhất (có số phần tử lớn nhất) và in ra màn hình dãy con đó.

CÁC NHÓM BÀI TOÁN NÂNG CAO

Nhóm các bài toán lập trình thường được phân loại theo thuật toán, theo cấu trúc dữ liệu dùng để giải. Ở bậc trung THCS các bài toán thường chỉ sử dụng các vòng lặp, sử dụng các kiến thức toán như UCLN, BCNN, số nguyên tố, sự chia hết của các số nguyên,…Các dạng toán thường gặp là:

- **Nhóm các bài toán số học:**
- **Nhóm các bài toán thao tác trên mảng một chiều.**
- **Các bài toán khác.**

1. Các bài toán số học.

Để giải các bài toán về số học giáo viên cần cho học sinh ứng dụng nhuần nhuyễn các kiến thức số học ở THCS chủ yếu dựa vào 2 phép toán DIV, MOD:

- **Thuật toán tìm UCLN của 2 số nguyên dương:**

Cho 2 số nguyên dương m, n. Tìm UCLN(m,n)

Học sinh thường dùng một trong 2 thuật toán

Thuật toán 1: Sử dụng phép trừ liên tục cho đến khi 2 số bằng nhau:

```
+ Nhập m, n
+ While m<>n do
If m>n then m:=m-n
Else n:=n-m;
+UCNN:=n;
```

Thuật toán này chạy chậm: Ví dụ với m=1000000000, n=1 phải chạy 1 tỷ phép toán.

Thuật toán 2: (Đối với HSG nên hướng các em sử dụng thuật toán này)

```
+ Nhập m, n
+ While n <> 0 do
Begin
r:= m mod n
m:=n;
n:=r;
End;
+UCLN:=m
```

Với thuật toán này khi m=1000000000, n=1, chỉ mất vài phép toán để tính được UCLN(m,n).

Để tìm UCLN của dãy a1, a2,…,an, cần lập hàm:

```
FUNCTION UCLN(a, b: longint): longint;
Khi đó + d:=a1;
+ for i:=2 to n do Begin b:=ai; d:=UCLN(d, b);End;
```

Thuật toán tìm BCNN của 2 số: Ta có BCNN(m,n)= (m * n) div UCLN(m, n).

Để tìm BCNN của dãy số nguyên dương a1, a2, …,an (n>=2)

```
BCNN:=a1; d:=a1;
For i:=2 to n do n do
Begin
   b:=ai
 BCNN:=BCNN*b;
 d:=UCLN(BCNN,b);
 BCNN:=BCNN div d ;
End ;
```

- **Kiểm tra số nguyên tố.**

Cho số nguyên P. Hỏi P có phải là số nguyên tố không?

Nhiều học sinh lập thuật toán đếm số ước của P. Nếu số ước của P là 2 thì kết luận P là số nguyên tố. Thuật toán đó không sai nhưng chậm.

```
FUNCTION Ngto(P:Integer): Boolean;
Var NT:Boolean;
 I:integer;
```

```
Begin
 NT:=P>1;
 For i:=2 to Trunc(Sqrt(P)) do
 If P mod I =0 then
 Begin NT:=False; Break End;
 Ngto:=NT;
End;
```

- **Đếm số chữ số của một số, tính tổng các chữ sô**

Bài toán : Nhập một số nguyên n. Số n có bao nhiêu chữ sô. Tính tổng các chữ số của n.

```
+TongCS:=0; SoCS:=0;
+ While n<>0 do
Begin
   Inc(SoCS);
   TongCS:=TongCS+ n mod 10;
   n := n div 10 ;
 End ;
```

- **Biểu diễn số tự nhiên n từ hệ đếm thập phân qua các hệ đếm khác và ngược lại**

Bài toán: Cho n là số nguyên dương biểu diễn trong hệ thập phân. Hãy biểu diễn n trong hệ đếm q – phân (1<q<10).

Dùng mảng : array[0..20] of byte ;

```
+ d:=-1;
+ While n<>0 do
 Begin
 Inc(d);
 a[d]:= n mod q
 n:=n div q;
 End; { lưu dãy chữ số q- phân theo thứ tự ngược}
+ for i:=d downto 0 do write(a[i]);
```

Một số bài toán số học.

***Bài 1.* Phân tích ra thừa số nguyên tố**

Cho số tự nhiên n (n>1). Hãy phân tích n thành tích các thừa số nguyên tố.

Ví dụ: Cho n=12 thì n=2.2.3, cho n=300 thì n=2.2.3.5.5

+ Phân tích: Chỉ cần duyệt qua các ước nguyên tố từ bé đến lớn rồi ghi ra.

+ Thuật toán:

```
If Ngto(n) then writeln(n)
 Else
 Begin
 m:=n;
 For p:=2 to n div 2 do
 If Ngto(p) then
 Begin
 While m mod p = 0 do write(p,'.');
 m:= m div p;
 End;
```

***Bài 2.* Rút gọn phân số.**

Cho phân số a/ b trong đó a nguyên còn b là số tự nhiên khác không. Hãy tìm 2 số c, d sao cho phân số c/ d tối giản và a/b=c/d.

+ Phân tích:

Đây là bài toán đơn giản, nhưng phải chú ý số a có thể âm.

+ Thuật toán:

```
Dau:=1; If a<0 then dau:=-1 ; a:=abs(a);
c:=a div UCLN(a,b); d:=b div UCLN(a, b);
Writeln(dau*c,'/ ', d);
```

Bài 3. **Giai thừa.**

Cho số tự nhiên n. P=n!. Hỏi:

a/ P có bao nhiêu chữ số không tận cùng.

b/ Số khác 0 tận cùng của P là chứ số nào.

+Phân tích :

a/ Số chữ số 0 cuối cùng chính là số ước bằng 10 của P! mà p!=a.10^k=a.2^k.5^k.

Do số ước 2 nhiều hơn ước 5, nên số 0 tân cùng là k. Vậy ta cần tính số lượng ước 5 của P !

+ Thuật toán: s5:=0;

. For m:=5 to n do

Begin

K:=m;

While k mod 5 = 0 do Begin Inc(s5); k:=k div 5 End;

End;

In ra kết quả : S5.

b/ Câu này dễ mắc sai lầm vừa nhân vừa xóa 0 cuối và chỉ giữ lại chữ số khác 0 cuối cùng.

+ Thuật toán: Giả sử So2, so5 là số lượng ước 2 và ước 5 của P.

S2:=0; S5:=0; P:=1;

For m:=2 to n do

Begin

K:=m;

While K mod 2 = 0 do Begin inc(so2), k:= k div 2 End;

While K mod 5 = 0 do Begin inc(so5), k:= k div 5 End;

P:=P* K mod 10.

End.

For i:= 1 to so2-so5 do Begin p:=p * 2 mod 10

In ra kết quả : P

Bài 4. **Tính tổng chữ số.**

Một quyển sách có n trang. Hỏi

a/ Tổng tất các chữ số đã ghi trên các trang sách.

b/ Mỗi chũa số xuất hiện bao nhiêu lân.

Thuật toán :

+ Lập hàm TongCS(K) để tính tổng các chữ số trong K

+ Dùng mảng a[0], a[1],…,a[9], trong đó a[i] số lần xuất hiện của chữ số i:

+ Sum:=0; fillchar(a, sizeof(a),0);

+ For m:=1 to n do

Begin

K:=m;

Sum:=Sum+TongCS(K);

While K <> 0 do

Begin

Inc(a[k mod 10])

K:=K div 10

End;

In ra kết quả: Sum, a[0],…, a[9]

Bài 5. Phân tích

Cho số tự nhiên n. Hãy phân tích n =A+B sao cho UCLN(A, B) là lớn nhất.

Với $2 \le n \le 10^9$.

Bài 6. **Số nguyên tố.**

Cho một số tự nhiên n>1 (1<n<1000001).. Tìm số k nguyên tố không vượt quá n trong các trường hợp sau:

a) k lớn nhất.

b) k có tổng các chữ số lớn nhất.

c) k là số đối xứng lớn nhất. (k là số đối xứng nếu đọc số đó từ trái qua phải hay từ phải qua trái đều như nhau. Ví dụ: các số 373, 3, 979…là các số đối xứng).

Câu nào không tìm được kết quả thì ghi số 0 thay thế.

Ví dụ: n=100 thì quả xuất ra màn hình a) 97; b) 89; c) 11

***Bài 7.* Số siêu nguyên tố.**

Số P gọi lầ số siêu nguyên tố, nếu nó nguyên tố và khi ta lần lượt bỏ các chữ số ở hàng đơn vị từ tái qua phải thì số mới nhận được vẫn là một số nguyên tố.

Ví dụ: 239 là số siêu nguyên tố vì 239 là số nguyên tố và 23, 2 cũng là các số nguyên tố, còn 431 là số nguyên tô, 43 cũng là số nguyên tố, nhưng 4 không nguyên tố nên 431 không phải là số siêu nguyên tố. Cho một số n (0<n<1000000) Hãy đếm số lượng các số siêu nguyên tố có n chữ số.

Lời giải

- **Phương án 1**: Duyệt toàn bộ:

+ Tìm số nhỏ nhất lớn nhất có n chữ số

Chẳng hạn với n=3 thì số nhỏ nhất và lớn nhất có 3 chữ số là: n1=100 và n2=999

+ Dùng một biến chạy p:

For p:= n1 to n2 do

If p thỏa mãn then tăng biến đếm.

+ Thuật toán này cũng được 40 % số test

- **Thuật toán tốt.**

```
+If n=1 thì a[1]:=2;[2]:=3; a[3]:=5, a[4]:=7; slg:=4
 + If n>1 then
Begin c[1]:=1; c[2]:=3; c[3]:=7; c[4]:=9;scs:=1;
 Repeat inc(scs); tg:=0; fillchar(b, sizeof(b),0)
 For k:=1 to slg do For g:=1 to 4 do
 Begin if NT(a[k]*10+c[g]) then
 Begin inc(tg); b[tg]:= NT(a[k]*10+c[g]) End
 End;
 a:=b;Slg:=tg; tg:=0;
Until scs=n.
End; {kết quả là Slg}
```

***Bài 8.* Phân số**

Phân số $\frac{p}{q}$ (p, q là 2 số nguyên dương) gọi là phân số đúng nếu $\frac{p}{q}$ <1. Còn $\frac{p}{q}$ gọi là phân số tối giản nếu UCLN(p,q)=1

Yêu cầu: Cho trước số nguyên n (3≤ n ≤ 50000000).

a) Tính số lượng các phân số đúng, tối giản $\frac{p}{q}$ mà p+q=n.

b) Tìm phân số đúng, tối giản $\frac{p}{q}$ lớn nhất mà p+q=n.

Bài 9: Viết chương trình nhập vào số nguyên dương N từ bàn phím. Sử dụng câu lệnh lặp FOR tính tổng sau: $S=1+\frac{1}{2}+\frac{1}{3}+...+\frac{1}{N}$ và đưa kết quả ra màn hình.

Bài 10. Viết chương trình nhập vào số nguyên dương N từ bàn phím. Sử dụng câu lệnh lặp FOR tính tổng sau: $S=1-\frac{1}{2}+\frac{1}{3}-\frac{1}{4}..+(-1)^{N+1}\frac{1}{N}$ và đưa kết quả ra màn hình.

Bài 11. Viết chương trình nhập vào số nguyên dương N từ bàn phím. Sử dụng câu lệnh lặp FOR tính tổng sau: $S=1-\frac{1}{2!}+\frac{1}{3!}-\frac{1}{4!}+...+(-1)^{N+1}\frac{1}{N!}$ và đưa kết quả ra màn hình.

Bài 12. Viết chương trình nhập từ bàn phím số nguyên dương N, tính tổng các ước thực sự của N và in ra màn hình. Ví dụ: N=6 thì tổng các ước là 1+2+3 =6; N=9 thì tổng các ước là 1+ 3 =4

Bài 13. Số N được gọi là số hoàn hảo nếu tổng các ước thực sự của N bằng chính nó. Viết chương trình nhập từ bàn phím số nguyên dương N, thông báo ra màn hình DUNG nếu N là số hoàn hảo, ngược lại thì thông báo KHONG.

Bài 14. Viết chương trình in ra màn hình các số hoàn hảo trong khoảng từ a đến b (với 1≤a<b). Với a, b được nhập từ bàn phím.

Bài 15. Lập chương trình tìm tất cả các số hoàn hảo nhỏ hơn số nguyên N (N >=10), in các số hoàn hảo và và các ước của số tìm được ra màn hình. Giá trị của N được nhập từ bàn phím.

Tương tự ta yêu cầu học sinh sao chép sửa lại chương trình bài tập 3 để được bài tập 4.

Bài 16. Hai số a và b được gọi là bạn của nhau nếu tổng các ước của a bằng b và ngược lại tổng các ước của b bằng a. Viết chương trình tìm các số bè bạn như trên trong khoảng từ m đến n (m < n, m và n nhập từ bàn phím).

Bài 17. Viết chương trình nhập vào 2 số a, b (1<a<b<10000). Thông báo ra màn hình số lượng các số chẵn, số lượng các số lẽ, số lượng các số hoàn hảo trong khoảng từ a đến b, với a, b được nhập từ bàn phím.

Bài 18: Nhập vào một số tự nhiên N (1 < N ≤ 65355). Hãy kiểm tra số N vừa nhập có phải là số nguyên tố hay không? Nếu đúng thì thông báo ra màn hình đây là số nguyên tố, ngược lại thì phân tích số N thành tích các thừa số nguyên tố.

Ví dụ:

Dữ liệu vào	Dữ liệu ra
Nhap 1 so tu nhien: 37 Nhap 1 so tu nhien: 30 Nhap 1 so tu nhien: 3456 Nhap 1 so tu nhien: 677	37 la so nguyen to 30 = 2. 3. 5 3456 = 2. 2. 2. 2. 2. 2. 2. 3. 3. 3 677 la so nguyen to

Bài 19. Số siêu nguyên tố là số nguyên tố mà khi bỏ một số tuỳ ý các chữ số bên phải của nó thì phần còn lại vẫn tạo thành một số nguyên tố.

Ví dụ 37337 là một số siêu nguyên tố có 5 chữ số vì 3733, 373, 37,3 cũng là các số nguyên tố.

Hãy viết chương trình đọc dữ liệu vào là một số nguyên N (0< N <10) từ bàn phím và đưa ra kết quả ra màn hình là các số siêu nguyên tố có N chữ số cùng số lượng của chúng.

Bài 20: Cho số nguyên N chẵn (nhập từ bàn phím), lập chương trình phân tích N thành tổng 2 số nguyên tố. Nếu có in ra các cách phân tích. VD: 6 = 3+ 3

2. Nhóm các bài toán mảng.

Trong phần này gv cần cung cấp cho học sinh một số kỹ năng cơ bản khi thao tác trên mảng như: Tìm phần tử MAX, MIN, sắp xếp đơn giàn, kiểm tra tinh đơn điệu của dãy, tìm kiếm trên dãy…

Bai 1. Trong một buổi sinh hoạt câu lạc bộ Tin học của Cung thiếu nhi Hà Nội, thầy giáo ra một bài toán như sau: Từ số hạng đầu tiên của dãy số Fibonacci (là dãy số có quy luật: số hạng thứ nhất và thứ hai bằng 1, từ số hạng thứ ba trở đi bằng tổng hai số hạng đứng ngay trước nó) thành lập dãy số mới gồm số bằng cách lần lượt thay mỗi số hạng bằng số dư của số hạng đó khi chia cho 100.

Ví dụ, với ta có 13 số hạng đầu tiên của dãy số Fibonacci là:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233

Dãy số mới nhận được sau khi thay là:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 44, 33

Để kiểm tra bài làm, yêu cầu các bạn trả lời hai thông tin sau:

- Số hạng thứ trong dãy mới là số nào?
- Có bao nhiêu giá trị khác nhau trong dãy số mới?

Nhập vào từ file FIBO.INP số nguyên dương n duy nhất.

Xuất ra màn hình:

+ Dòng 1: in ra số hạng thứ trong dãy mới.

+ Dòng 2: in ra số lượng các giá trị khác nhau trong dãy mới.

Ví dụ:

Nhập	
Kết quả ra màn hình	***Giải thích***
33	Số hạng thứ 13 trong dãy mới là số 33
12	Có 12 giá trị khác nhau trong dãy số mới

Bài 2. **Tổng max.**

Cho dãy n số nguyên dương a_1, a_2,…a_n. Hãy tìm 2 số a_i, a_j, sao cho $i \neq j$ và a_i+a_j đạt max.

Bài 3. Viết chương trình nhập vào từ bàn phím số nguyên N và mảng A gồm N phần tử. Thông báo ra màn hình các số chẵn có trong mảng đã nhập và số lượng của chúng.

Bài 4. Viết chương trình nhập vào từ bàn phím số nguyên N và mảng A gồm N phần tử. Thông báo ra màn hình các số lẽ có trong mảng đã nhập và số lượng của chúng.

Bài 5. Viết chương trình nhập vào từ bàn phím số nguyên N, số K và mảng A gồm N phần tử. Thông báo ra màn hình các số chia hết cho K có trong mảng đã nhập và số lượng của chúng.

Bài 6. Viết chương trình nhập vào từ bàn phím số nguyên N và mảng A gồm N phần tử. Thông báo ra màn hình các số nguyên tố có trong mảng đã nhập và số lượng của chúng.

Bài 7. Viết chương trình nhập vào từ bàn phím số nguyên N và mảng A gồm N phần tử. Thông báo ra màn hình các số hoàn hảo có trong mảng đã nhập và số lượng của chúng.

Bài 8. Viết chương trình vào từ bàn phím số nguyên N và mảng A gồm N phần tử. In ra màn hình mảng A sau khi đã sắp xếp các phần tử theo thứ tự tăng dần.

Bài 9. Viết chương trình vào từ bàn phím số nguyên N và mảng A gồm N phần tử. In ra màn hình mảng A sau khi đã sắp xếp các phần tử theo thứ tự giảm dần.

Bài 10. Viết chương trình nhập vào từ bàn phím mảng A gồm N phần tử là số nguyên gồm 3 loại số : Loại 1 : các số vừa chia hết cho 3 vừa lẻ, loại 3 : các số vừa chia hết cho 3 vừa chẵn, loại 2 : các số còn lại. Yêu cầu hãy xếp các số loại 1 lên đầu dãy, các số loại 3 xuống cuối dãy, các số loại 2 ở giữa dãy. Trước hết xếp gọn hết các số loại 1, sau đó xếp đồng thời các số loại 2 và 3. Đưa ra màn hình dãy đã được sắp xếp.

Dữ liệu vào
N=8 88 29 81 12 42 35 62 83

Dữ liệu ra
88 29 81 12 42 35 62 83 81 29 88 62 83 35 12 42

Bài 11. Viết chương trình nhập vào từ bàn phím mảng một chiều A gồm M phần tử, mảng một chiều B gồm N phần tử, sắp xếp 2 dãy A và B tăng dần sau đó trộn 2 dãy thành dãy C sao cho dãy C cũng là dãy đã được sắp xếp tăng dần. In mảng A, B, C ra màn hình.

Bài 12. Nhập vào một mảng các số nguyên.

a/ Xếp lại mảng đó theo thứ tự giảm dần.

b/ Nhập vào một số nguyên từ bàn phím. Chèn số đó vào mảng sao cho mảng vẫn có thứ tự giảm dần. (không được xếp lại mảng)

Bài 13. Nhập mảng gồm N số nguyên từ bàn phím. Sau đó nhập số nguyên X. Hãy loại bỏ ở dãy các phần tử bằng X. In dãy ra màn hình

VD : Dãy: 1 2 6 5 4 3 6 7 6 với X = 6 => Dãy in ra: 1 2 5 4 3 7

BỘ ĐỀ THAM KHẢO

Đề 1: ***Tổng quan bài thi:***

	Tên bài	Tên chương trình	Dữ liệu vào	Dữ liệu ra	Điểm
Câu 1	*Diện tích hình thang.*	Câu1.pas	*Bàn phím*	*Màn hình*	3,0
Câu 2	*Hóa đơn.*	Câu2.pas	*Bàn phím*	*Màn hình*	3,0
Câu 3	*Lãi suất.*	Câu3.pas	*Bàn phím*	*Màn hình*	1,5
Câu 4	*Tổng của k số.*	Câu4.pas	*Bàn phím*	*Màn hình*	1,5
Câu 5	*Xếp hạng.*	Câu5.pas	*Bàn phím*	*Màn hình*	1,0

Thí sinh dùng ngôn ngữ lập trình Free Pascal (hoặc Turbo Pascal), hãy viết chương trình cho các bài sau đây:

Câu 1 (3,0 *điểm*). *Diện tích hình thang.*

Tính diện tích *S* của hình thang biết độ dài đáy lớn là *a*, độ dài đáy bé là *b* và độ dài đường cao là *h*, trong đó *a, b* và *h* được nhập từ bàn phím. In kết quả ra màn hình.

(Biết diện tích hình thang được tính theo công thức $s = (a+b).\frac{h}{2}$*).*

Câu 2 (3,0 *điểm*). *Hóa đơn.*

Nhập vào số dương *N* là số dung lượng mà khách hàng đã sử dụng dịch vụ Internet trong một tháng. Hãy tính hóa đơn tiền *(đơn vị đồng)* mà khách hàng phải thanh toán cho nhà cung cấp dịch vụ, In kết quả ra màn hình, biết *thành tiền=N.Đơn giá.* Trong đó:

- Nếu $N \le 50$ Mb thì hóa đơn được tính với đơn giá là 450.
- Nếu $N>50$ Mb thì hóa đơn của 50 Mb đầu được tính với đơn giá là 450, còn $(N-50)$ Mb còn lại được tính với đơn giá 500.

Câu 3 (1,5 *điểm*). *Lãi suất.*

Bạn Lan gửi một số tiền là *A* vào ngân hàng với lãi suất 15% trên một tháng. Hỏi sau bao nhiêu tháng thì bạn Lan rút được số tiền là *B*? Trong đó *A* và *B* là các giá trị được nhập từ bàn phím. In kết quả ra màn hình.

Biết tiền gửi của tháng sau được tính bằng tiền gốc và tiền lãi của tháng trước: A=A+0,15.A (đơn vị nghìn đồng).

Câu 4 (1,5 *điểm*). *Tổng của k số.*

Cho dãy *A* gồm *N* phần tử $A_1, A_2, \ldots, A_N$ và số nguyên k $(1 \le k \le N)$. Hãy tính tổng của *k* phần tử đầu tiên trong dãy. In kết quả ra màn hình.

Ví dụ: Cho *N*=6, dãy *A* gồm 5, 1, 3, 10, 9, 5 và *k*=4 thì tổng của *k* phần tử đầu tiên trong dãy là 19.

Câu 5 (1,0 *điểm*). *Xếp hạng.*

Cho dãy *A* gồm *N* phần tử $A_1, A_2, \ldots, A_N$. Hãy xếp thứ hạng cho các phần tử của dãy số trên biết hai số bằng nhau có cùng thứ hạng, số lớn nhất xếp hạng 1, số lớn thứ 2 xếp hạng 2,... In kết quả ra màn hình.

Ví dụ: Cho *N*= 6, dãy *A* gồm các số 12, 9, 20, 15, 9, 20, 5 vậy kết quả xếp hạng là:

Kết quả: Dãy số: 12 9 20 15 9 20 5
Xếp hạng: 3 4 1 2 4 1 5

-------- Hết -------

Đề 2: **TỔNG QUAN BÀI THI**

Câu	Tên bài	Tên chương trình	Dữ liệu vào	Dữ liệu ra	Điểm
1	*Tính diện tích hình tròn*	Cau1.pas	*Bàn phím*	*Màn hình*	3,0
2	*Tìm Max*	Cau2.pas	*Bàn phím*	*Màn hình*	3,0
3	*Dãy số*	Cau3.pas	*Bàn phím*	*Màn hình*	1,5
4	*Tính tổng các số chẵn*	Cau4.pas	*Bàn phím*	*Màn hình*	1,5
5	*Số nhỏ thứ nhì*	Cau5.pas	*Bàn phím*	*Màn hình*	1,0

Thí sinh sử dụng ngôn ngữ lập trình, hãy lập trình cho các bài toán sau:

Câu 1: (3,0 điểm). *Tính diện tích hình tròn*

Lập trình tính diện tích hình tròn có chiều dài bán kính là r (r được nhập vào từ bàn phím với r > 0). Xuất kết quả ra màn hình. *(Biết diện tích hình tròn S= 3.14*r*r)*

Câu 2: (3,0 điểm). *Tìm Max*

Lập trình tìm số lớn nhất trong 4 số a, b, c, d được nhập vào từ bàn phím. Xuất kết quả ra màn hình.

Ví dụ: *Nhập: a = 3; b = 8; c = - 10; d = 5*

Kết quả xuất ra màn hình: "Max la: 8"

Câu 3: (1,5 điểm). *Dãy số*

Lập trình xuất kết quả ra màn hình các số chẵn từ 2 đến N, nếu N là số chẵn. Xuất kết quả ra màn hình các số lẻ từ 1 đến N, nếu N là số lẻ. Với N là một số nguyên dương được nhập vào từ bàn phím (N≤100).

Ví dụ: N = 10 *Kết quả xuất ra màn hình*: 2 4 6 8 10

N = 15 *Kết quả xuất ra màn hình*: 1 3 5 7 9 11 13 15

Câu 4: (1,5 *điểm*). *Tính tổng các số chẵn*

Lập trình tìm tổng các số chẵn trong dãy số gồm *N* số nguyên A_1, A_2, ..., A_N. Dãy số A_1, A_2, ..., A_N được nhập từ bàn phím (N≤100). Xuất kết quả ra màn hình.

Ví dụ: *Nhập* N = 5, *dãy số:* 4; 9; 15; 7; -12

Kết quả xuất ra màn hình: - 8

Câu 5: (1,0 điểm). *Số nhỏ thứ nhì*

Lập trình tìm số nhỏ thứ nhì trong dãy A_1, A_2, ..., A_N , gồm N số nguyên được nhập vào từ bàn phím (N≤100), xuất kết quả ra màn hình.

Ví dụ: *Nhập* N = 5; *dãy số:* 20; 10; 9; 9; 8

Kết quả xuất ra màn hình: "So nho thu nhi: 9"

Nhập N = 8; *dãy số*: 101; 50; 25; 55; 50; 90; 190; 25

Kết quả xuất ra màn hình: "So nho thu nhi: 50"

----- Hết -----